

desired rate code $r(t)$ would be equal to $R(4)$. Next RC compares $r(t)$ with $r(t-1)$ stored in CR. If $r(t)$ with $r(t-1)$ are equal then no new rate code needs to be sent to fabric 202. If, for example at time t , $r(t-1)$ in CR is $R(3)$ and $r(t)$ is $R(4)$ then $g(t)$ just moved above the $Q(4)$ boundary and it is time to send the $R(4)$ code to fabric 202. Conversely, suppose that $r(t-1)$ is $R(5)$; in this case $q(t)$ just slipped below $Q(5)$ and it is time to send the $R(4)$ code to the fabric. After the $r(t)$ and $r(t-1)$ comparisons are done the value of $r(t)$ is stored into CR and t advances. This embodiment is described as being employed in a line card. This embodiment should not be limited thereto. Rather, this embodiment may find application in the switching fabric to moderate the rate at which a line card transmits data to the switching fabric.

The formatter 252 of FIG. 2b may also include circuitry or an algorithm which prevents frequent transmission of control codes to fabric 202 when the contents of buffer 250 oscillates around one of the predefined quantities $Q(1) - Q(n)$. With $Q(1) - Q(n)$ fixed -- suppose that the $q(t)$ increases to become one greater than $Q(2)$. In this situation, $R(2)$ is sent to the fabric 202. In the next clock cycle, buffer 250 empties by one line so that $q(t)$ is no longer greater than $Q(2)$. In response, $R(1)$ is sent to fabric 202. At the next clock cycle, one line is stored in the buffer, so $q(t)$ exceeds $Q(2)$; and $R(2)$ is again sent to the fabric 202. If $q(t)$ is oscillating around the value $Q(2)$, rate codes will be frequently sent to fabric 202.

One way to avoid frequently sending rate codes to fabric 202 due to $q(t)$ oscillation around one of the programmed quantities $Q(1)$ through $Q(n)$, is to dynamically modify the value $Q(x)$ used for comparison with $q(t)$ as the $Q(x)$ threshold is crossed by $q(t)$ from below. With reference to the embodiment in which line card 204 includes a formatter F, a data buffer B that receives data from the fabric 202, a first set of quantity comparators $QC(1)$ through $QC(n)$, a clocked register CR for storing a rate code $r(t-1)$, and a rate comparator RC, suppose $q(t)$ exceeds $Q(2)$. The rate code $R(2)$ is sent to the fabric 202 as before, but now the value of $Q(2)$ used in the comparators (and only the value of $Q(2)$, not any other Q values) is changed by an amount Y set in a programmable register PR. At the next clock cycle the comparator $QC(2)$ compares $q(t)$ with $[Q(2) - Y]$ instead of $Q(2)$. For example, suppose $Q(2)$ is initially programmed to 500, and Y is set to 25. When $q(t)$ exceeds 500 the rate code $R(2)$ is sent to fabric 202, but only when $q(t)$ drops below 475 (which is $Q(2) - Y$) will $R(1)$ be sent to fabric 202. In this manner small oscillations of $q(t)$ around a $Q(x)$ value will not cause different rate codes to be sent frequently to

fabric 202. Note that the other $Q(1)$ through $Q(n)$ values other than the $Q(x)$ just crossed are unaffected. If, for example, $Q(3)$ is initially programmed to 580. If $q(t)$ exceeds $Q(2)=500$, $Q(2)$ is replaced with $Q(2)-Y$, and the $R(2)$ code is sent, then the $R(3)$ code will not be sent until $q(t)$ exceeds 580, not when it exceeds 555. This method works in both directions. If $q(t)$ rises from 0 to exceed 500, the $q(t)$ comparison for that comparator is against the level 475. However once $q(t)$ drops below 475, $Q(2)=475$ is replaced with 500 to avoid frequently sending different rate codes if the $q(t)$ value now hovers around 475.

Although the present invention have been described in connection with several embodiments, the invention is not intended to be limited to the specific forms set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents as can be reasonably included with in the spirit and scope of the invention as defined by the appended claims.